

**INVENTORS:** Antonino LA MALFA  
Salvatore POLI  
Paolino SCHILLACI

**STRUCTURE FOR UPDATING A BLOCK OF MEMORY CELLS IN A  
FLASH MEMORY DEVICE WITH ERASE AND PROGRAM  
OPERATION REDUCTION**

**Cross-Reference To Related Applications**

[0001] This application is based upon and claims priority from prior Italian Patent Application No. MI2002A002192, filed on October 16, 2002 the entire disclosure of which is herein incorporated by reference.

**Field of the Invention**

The present invention generally relates to flash memory devices and more particularly to a structure and method for updating a block of memory cells in a flash memory device.

**Background of the Invention**

[0002] A flash memory device is a particular type of E<sup>2</sup>PROM (Electrically Erasable and Programmable Read-Only Memory), which is used to store information that must be preserved even when a power supply is off; the flash memory is erased in blocks instead of one bit at a time. This results in a very simple structure of the flash memory, which can be manufactured at low cost and with high density. As a consequence, the flash memory is well suited to a number of end-product applications, such as Personal Computers (PCs), cellular phones, automotive devices, digital still cameras, and the like.

[0003] The flash memory includes a matrix of memory cells; typically, the matrix is partitioned into several sectors, which may be erased individually. An additional miniature matrix is often used to store configuration information

**EXPRESS MAIL LABEL NO. EV343427125US**

in a permanent way. For example, the miniature matrix implements a (non-volatile) protection register for the sectors of the matrix.

[0004] The configuration of the protection register identifies which sectors are locked. A reading operation may be performed on every sector; however, an erase operation or a program operation is only allowed on the unlocked sectors. This mechanism makes it possible to protect the matrix from an undesired erasing and/or programming of the memory cells, thereby preventing any loss of data stored in the flash memory.

[0005] The configuration of the protection register is updated overriding its content with a new value. This operation involves the erasure of the whole protection register; the memory cells required to reach the new value are then programmed.

[0006] A drawback of the solution described above is that every updating of the protection register necessitates both an erase operation and a program operation. As a consequence, the update operation is (relatively) slow. During this operation all the functional units of the flash memory are busy, so that no other operation can be performed on the matrix.

[0007] Moreover, the memory cells of the protection register are subjected to an electrical stress during each erase and program operation; after repeated cycles of updating, this may cause loss of data stored in the protection register.

[0008] Accordingly what is needed is a structure to overcome the problems encountered in the prior art for updating a block of memory cells in a flash memory device.

**Summary of the Invention**

[0009] The present invention overcomes the above-mentioned drawbacks by providing a structure for updating a block of memory cells in a flash memory device, the memory cells storing a current value, wherein the structure includes means for receiving a new value to be written on the memory cells, erasing means for erasing the block of memory cells simultaneously, and programming means for programming the memory cells individually; the structure further includes control means for enabling the erasing means and for enabling the programming means according to a comparison between the new value and the current value.

[0010] Moreover, the present invention provides a flash memory device including this structure. A corresponding method of updating a block of memory cells in a flash memory device is also encompassed.

**Brief Description of the Drawing**

[0011] The subject matter, which is regarded as the invention, is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other features, and advantages of the invention will be made clear by the following description of a preferred embodiment thereof, given purely by way of a non-restrictive indication, with reference to the attached figures, in which:

[0012] FIG. 1 is a schematic block diagram of a flash memory in which the structure of the invention can be used;

[0013] FIGs. 2a and 2b show different circuit schemes implementing the proposed structure.

**Description Of The Preferred Embodiments**

**EXPRESS MAIL LABEL NO. EV343427125US**

[0014] It should be understood that these embodiments are only examples of the many advantageous uses of the innovative teachings herein. In general, statements made in the specification of the present application do not necessarily limit any of the various claimed inventions. Moreover, some statements may apply to some inventive features but not to others. In general, unless otherwise indicated, singular elements may be in the plural and vice versa with no loss of generality.

[0015] With reference in particular to FIG. 1, a writing section of a flash memory device 100 is illustrated. The flash memory 100 is integrated in a chip of semiconductor material. A matrix 105 of memory cells (for example, with a so-called NOR architecture) is used to store program code or data. The matrix 105 is segmented into a plurality of sectors (for example, 16 sectors from 0 to 15) made in respective insulated wells of the chip.

[0016] Each memory cell consists of a floating gate MOS transistor. The memory cell in a non-programmed (or erased) condition features a low threshold voltage (associated with a logic value 1). The memory cell is programmed by injecting electric charge into its floating gate; in this condition, the memory cell features a high threshold voltage (associated with a logic value 0). Conversely, the memory cell is erased by removing the electric charge accumulated in its floating gate; all the memory cells of a sector must be erased at the same time.

[0017] A miniature matrix 110 (close to the matrix 105) is used as a non-volatile protection register for the sectors of the matrix 100. The configuration of the protection register 110 identifies which sectors are locked, so that a program operation or an erase operation cannot be performed. For example, the protection register 110 consists of 3 memory cells (each one storing a bit); the  $2^3=8$  combinations of the bits stored in the memory cells define the following protection conditions:

Protection register			Locked sectors
0	0	0	None
0	0	1	Sector 15

**EXPRESS MAIL LABEL NO. EV343427125US**

0	1	0	Sectors 14-15
0	1	1	Sectors 12-15
1	0	0	Sectors 8-16
1	0	1	All (Sectors 0-15)
1	1	0	All (Sectors 0-15)
1	1	1	All (Sectors 0-15)

[0018] The information stored in the protection register 110 is updated erasing and/or programming its memory cells; all the memory cells of the protection register 110 must be erased at the same time.

[0019] A column decoder 115c and a row decoder 115r select the memory cells of either the matrix 105 or the protection register 110. A bank of sense amplifiers 120 is used to read the values stored in the selected memory cells; a bank of program loads 125 is instead used to program the selected memory cells. The sense amplifiers 120 also interface with a volatile register (or latch) 127, which operates as a buffer for the protection information read from the register 110 (at the power-on of the flash memory 100).

[0020] The flash memory 100 has a serial interface for an external bus conforming to the SPI specification. Particularly, an input buffer 130 latches information received from the external bus. The input information is then provided to a state machine 135, which substantially operates as a series-to-parallel converter. The input information consists of instructions, addresses of memory cells, or data to be written onto selected memory cells.

[0021] Any address received from the external bus is stored in a latch 140, and it is then used to drive the column decoder 115c and the row decoder 115r. Likewise, any data is stored in a further latch 145. The data stored in the latch 145 is used to select the program loads 125 to be actuated (for the memory cells that need to be programmed).

[0022] Any instruction is instead provided to a decoder 150, which also receives the address stored in the latch 140 and the protection information stored in the latch 127. The decoder 150 interprets the instruction; moreover, it verifies whether the instruction involves an erase operation or a program operation on memory cells of a locked sector. Any instruction attempting to

## EXPRESS MAIL LABEL NO. EV343427125US

erase or to program a locked sector is aborted; otherwise, the decoder 150 actuates a controller 155 accordingly. The controller 155 generates a sequence of control signals (denoted as a whole with Sc), which cause execution of the operations required by the instruction.

[0023] A specific instruction is used to update the content of the protection register 110. A new value to be written on the protection register 110 (stored in the latch 145) and its current value (stored in the latch 127) are provided to a control logic 160. As described in detail in the following, the control logic 160 outputs an erase-enabling signal EE and a program-enabling signal EP for the controller 155. The signal EE is asserted when the updating of the protection register 110 requires an erase operation; the signal EP is instead asserted when the updating of the protection register 110 requires a program operation.

[0024] In the flash memories known in the art, any updating of the protection register 110 involves the overriding of its content with the new value. As a consequence, the whole protection register 110 is at first erased; the memory cells required to reach the new value are then programmed.

[0025] However, this procedure is not optimal. In fact, the erase operation and the program operation are not always required to reach the new value from the current value. The operations actually necessary for every combination current value/new value (64 cases) are shown in the following table:

Current value			New value			Erase	Program
0	0	0	0	0	0	No	No
0	0	0	0	0	1	Yes	Yes
0	0	0	0	1	0	Yes	Yes
0	0	0	0	1	1	Yes	Yes
0	0	0	1	0	0	Yes	Yes
0	0	0	1	0	1	Yes	Yes
0	0	0	1	1	0	Yes	Yes
0	0	0	1	1	1	Yes	No
0	0	1	0	0	0	No	Yes
0	0	1	0	0	1	No	No
0	0	1	0	1	0	Yes	Yes
0	0	1	0	1	1	Yes	Yes

**EXPRESS MAIL LABEL NO. EV343427125US**

0	0	1	1	0	0	Yes	Yes
0	0	1	1	0	1	Yes	Yes
0	0	1	1	1	0	Yes	Yes
0	0	1	1	1	1	Yes	No
0	1	0	0	0	0	No	Yes
0	1	0	0	0	1	Yes	Yes
0	1	0	0	1	0	No	No
0	1	0	0	1	1	Yes	Yes
0	1	0	1	0	0	Yes	Yes
0	1	0	1	0	1	Yes	Yes
0	1	0	1	1	0	Yes	Yes
0	1	0	1	1	1	Yes	No
0	1	1	0	0	0	No	Yes
0	1	1	0	0	1	No	Yes
0	1	1	0	1	0	No	Yes
0	1	1	0	1	1	No	No
0	1	1	1	0	0	Yes	Yes
0	1	1	1	0	1	Yes	Yes
0	1	1	1	1	0	Yes	Yes
0	1	1	1	1	1	Yes	No
1	0	0	0	0	0	No	Yes
1	0	0	0	0	1	Yes	Yes
1	0	0	0	1	0	Yes	Yes
1	0	0	0	1	1	Yes	Yes
1	0	0	1	0	0	No	No
1	0	0	1	0	1	Yes	Yes
1	0	0	1	1	0	Yes	Yes
1	0	0	1	1	1	Yes	No
1	0	1	0	0	0	No	Yes
1	0	1	0	0	1	No	Yes
1	0	1	0	1	0	Yes	Yes
1	0	1	0	1	1	Yes	Yes
1	0	1	1	0	0	No	Yes
1	0	1	1	0	1	No	No
1	0	1	1	1	0	Yes	Yes
1	0	1	1	1	1	Yes	No
1	1	0	0	0	0	No	Yes
1	1	0	0	0	1	Yes	Yes
1	1	0	0	1	0	No	Yes
1	1	0	0	1	1	Yes	Yes
1	1	0	1	0	0	No	Yes
1	1	0	1	0	1	Yes	Yes
1	1	0	1	1	0	No	No
1	1	0	1	1	1	Yes	No
1	1	1	0	0	0	No	Yes
1	1	1	0	0	1	No	Yes
1	1	1	0	1	0	No	Yes

**EXPRESS MAIL LABEL NO. EV343427125US**

1	1	1	0	1	1	No	Yes
1	1	1	1	0	0	No	Yes
1	1	1	1	0	1	No	Yes
1	1	1	1	1	0	No	Yes
1	1	1	1	1	1	No	No

[0026] As can be seen, in 8 cases out of 64 (about 12%) no operation must be performed (since the new value is already stored in the protection register 110). Only the erase operation is required in 19 cases out of 64 (about 30%), and only the program operation is required in 7 cases out of 64 (about 11%). Both the erase operation and the program operation are required in the remaining 47 cases out of 64 (about 47%).

[0027] Therefore, assuming that the pairs current value/new value are uniformly distributed among the 64 possible combinations, it results that (on the average) an erase operation is performed even when it is not necessary in the 23% of the cases (12% + 11%); likewise, a program operation is performed even when it is not necessary in the 42% of the cases (12% + 30%).

[0028] In sharp contrast to the solutions known in the art, the structure described in the following compares the new value to be written on the protection register 110 with its current value; the protection register 110 is then erased and/or programmed only when it is necessary (according to the result of the comparison).

[0029] However, the concepts of the present invention are also applicable when the flash memory has another architecture, when the matrix is partitioned in a different way, or when the protection register has another structure; similar considerations apply if equivalent information is stored in the protection register, if the flash memory has a different interface (even of the parallel type), and the like.

[0030] Alternatively, the proposed structure is used to update other non-volatile registers of the flash memory (for example, storing configuration information). Moreover, the same structure finds application in a so-called



page flash as well, wherein small blocks of memory cells (for example, storing 4-16 bits) may be erased individually; in this case, the proposed solution is used to update every block of memory cells of the matrix.

[0031] In an embodiment of the present invention, as shown in FIG. 2a, the control logic is implemented with a combinatorial circuit 200a.

[0032] Particularly, each bit  $Pn_i$  (with  $i=0..2$ ) of the new value to be written on the protection register is supplied to a NOT logic gate 205<sub>i</sub>, which outputs a corresponding inverted bit  $\underline{Pn}_i$ . An OR logic gate 210<sub>i</sub> is provided for each bit; the OR gate 210<sub>i</sub> is input the inverted bit  $\underline{Pn}_i$  and a corresponding bit  $Pc_i$  of the current value stored in the protection register. The output terminals of the OR gates 210<sub>i</sub> are connected to respective input terminals of a NAND logic gate 215, which generates the erase-enabling signal EE.

[0033] Moreover, each inverted bit  $\underline{Pn}_i$  and each bit  $Pc_i$  are AND-ed by a corresponding logic gate 220<sub>i</sub>. The signals provided by the AND gates 220<sub>i</sub> are applied to respective input terminals of a NOR logic gate 225. The output signal of the NOR gate 225 is inverted by a NOT logic gate 230, which generates the program-enabling signal EP.

[0034] Whenever the content of the protection register must be updated, its current value (stored in the protection latch) is compared with the desired new value (stored in the data latch). As a consequence, the erase-enabling signal EE is asserted when an erase operation is required to reach the new value from the current value. In detail, if a memory cell is programmed in the current value ( $Pc_i=0$ ) and erased in the new value ( $Pn_i=1$ ), the output signal of the corresponding OR gate 210<sub>i</sub> is deasserted. When this condition is satisfied for one or more bits, the erase-enabling signal EE output by the NAND gate 215 is asserted. In response thereto, the controller erases all the memory cells of the protection register at the same time; the protection register is then read to verify the completion of the erase operation, and the content of the protection latch is updated accordingly ( $Pc_2=1$ ,  $Pc_1=1$  and

**EXPRESS MAIL LABEL NO. EV343427125US**

$P_{C0}=1$ ). Conversely, the erase-enabling signal EE remains deasserted, and no operation is performed on the protection register.

[0035] The up-to-date current value of the protection register (after the possible erase operation) is compared again with the desired new value. As a consequence, the program-enabling signal EP is asserted when a program operation is required to reach the new value from the (up-to-date) current value. In detail, if a memory cell is erased in the current value ( $P_{Ci}=1$ ) and programmed in the new value ( $P_{Ni}=0$ ), the output signal of the corresponding AND gate 220<sub>i</sub> is asserted. When this condition is satisfied for one or more bits, the output signal of the NOR gate 225 is deasserted; therefore, the program erase-enabling signal EP output by the NOT gate 230 is asserted. In response thereto, the controller programs the cells of the protection register required to reach the new value; the protection register is then read to verify the completion of the program operation, and the content of the protection latch is updated accordingly. Conversely, the program-enabling signal EP remains deasserted, and no operation is performed on the protection register (since it already stores the desired new value).

[0036] Considering now FIG. 2b, an alternative embodiment of the present invention is illustrated. In this case, the control logic is implemented with a different combinatorial circuit 200b (the elements corresponding to the ones shown in FIG. 2a are denoted with the same references, and their explanation is omitted for the sake of simplicity).

[0037] Particularly, the erase-enabling signal EE is generated by the logic gates 205<sub>i</sub>, 210<sub>i</sub>, 215 as described above. The logic gates 205<sub>i</sub>, 220<sub>i</sub>, 225 instead generate an internal signal denoted with Ea (the internal signal Ea is underlined to specify that it is deasserted at logic value 1 and asserted at logic value 0). All the inverted bits P<sub>n0</sub>-P<sub>n2</sub> are further OR-ed by a logic gate 235. A NAND logic gate 240 is input the signal generated by the OR gate 235 and the erase-enabling signal EE (generated by the NAND gate 215); the NAND gate 240 outputs a further internal signal denoted with Eb. The internal signal Ea (from the NOR gate 225) and the internal signal Eb (from

**EXPRESS MAIL LABEL NO. EV343427125US**

the NAND gate 240) are provided to respective input terminals of a NAND logic gate 245; the NAND gate 245 generates the program-enabling signal EP.

[0038] Whenever the content of the protection register must be updated, its current value (stored in the protection latch) is compared with the desired new value (stored in the data latch). As described above, the erase-enabling signal EE is asserted when an erase operation is required to reach the new value from the current value. In response thereto, the controller erases all the memory cells of the protection register at the same time; however, the content of the protection latch is not updated. Conversely, the erase-enabling signal EE remains deasserted, and no operation is performed on the protection register.

[0039] The (original) current value of the protection register is compared again with the desired new value. As a consequence, the program-enabling signal EP is asserted when a program operation is required to reach the new value from the current value; at the same time, the program-enabling signal EP is also asserted when the protection register has been erased but one or more memory cells must be programmed in the new value.

[0040] In detail, if a memory cell is erased in the current value ( $P_{Ci}=1$ ) and programmed in the new value ( $P_{Ni}=0$ ), the output signal of the corresponding AND gate 220<sub>i</sub> is asserted. When this condition is satisfied for one or more bits, the internal signal Ea output by the NOR gate 225 is asserted (0). Otherwise, the internal signal Ea remains deasserted (1). Moreover, if one or more of the memory cells are programmed in the new value ( $P_{Ni}=0$ ), the output signal of the OR gate 235 is asserted. If at the same time the protection register has been erased ( $EE=1$ ), the internal signal Eb output by the NAND gate 240 is asserted (0). Otherwise, the internal signal Eb remains deasserted (1). If the internal signal Ea or the internal signal Eb are asserted, the program-enabling signal EP is asserted. In response thereto, the cells of the protection register required to reach the new value are programmed, and the protection latch is updated accordingly. Conversely, the program-

**EXPRESS MAIL LABEL NO. EV343427125US**

enabling signal EP remains deasserted, and no operation is performed on the protection register.

[0041] However, the concepts of the present invention are also applicable when the control logic is implemented with an equivalent combinatorial circuit, when the logic gates are interconnected in another way, when different logic gates are used, and the like.

[0042] More generally, the present invention proposes a structure for updating a block of memory cells (storing a current value) in a flash memory device. The structure includes means for receiving a new value to be written on the memory cells. Erasing means are provided for erasing the block of memory cells simultaneously; programming means are instead provided for programming the memory cells individually. The structure of the invention further includes control means for enabling the erasing means and for enabling the programming means according to a comparison between the new value and the current value.

[0043] In the proposed solution, the block of memory cells is erased and/or programmed only when it is necessary (according to the result of the comparison between the value currently stored and the new value to be written).

[0044] Therefore, in several cases the updating of the memory cells only involves an erase operation or a program operation. This strongly increases the speed of the whole update operation.

[0045] The structure of the invention avoids the involvement of the functional units required to erase and/or to program the memory cells when they are not necessary (thereby leaving the flash memory device available for other operations).

[0046] Moreover, the devised solution limits the electrical stress suffered by the memory cells during each update operation; in this way, the risk of loss of data is strongly reduced.

**EXPRESS MAIL LABEL NO. EV343427125US**

[0047] The preferred embodiment of the invention described above offers further advantages.

[0048] Particularly, the structure is implemented with a combinatorial circuit.

[0049] This solution is very simple, but at the same time effective.

[0050] As a further enhancement, the current value of the protection register is stored in a corresponding latch.

[0051] The proposed feature removes the need to perform any reading operation on the protection register for comparing its current value with the desired new value.

[0052] Typically, the block of memory cells consists of a protection register (even if different applications are not excluded).

[0053] Alternatively, equivalent control means is used (even of the sequential type), no latch is provided for the information stored in the protection register, or the proposed structure is used to update other blocks of memory cells.

[0054] In a particular embodiment of the invention, the control logic compares the new value of the protection register with its up-to-date current value (after a possible erase operation), in order to determine whether a program operation is required.

[0055] This solution simplifies the implementation of the control logic (even if the duration of the update operation is slightly increased).

[0056] In a different embodiment of the invention, the control logic always compares the new value of the protection register with its original current value.

[0057] This solution is faster, since no reading of the up-to-date current value of the protection register is required (even if the complexity of the control logic is slightly increased).

**EXPRESS MAIL LABEL NO. EV343427125US**

[0058] In both cases, a suggested choice for the implementation of the control logic consists of a set of logic gates.

[0059] The proposed structure is very simple and fast.

[0060] However, the solution according to the present invention leads itself to be put into practice with a different mechanism for comparing the value currently stored in the protection register with the new value to be written, or even implementing the control logic with another circuit.

[0061] Naturally, in order to satisfy local and specific requirements, a person skilled in the art may apply to the solution described above many modifications and alterations all of which, however, are included within the scope of protection of the invention as defined by the following claims.

[0062] Although a specific embodiment of the invention has been disclosed, it will be understood by those having skill in the art that changes can be made to this specific embodiment without departing from the spirit and scope of the invention. The scope of the invention is not to be restricted, therefore, to the specific embodiment, and it is intended that the appended claims cover any and all such applications, modifications, and embodiments within the scope of the present invention.

[0063] What is claimed is: